

DESCRIPTION

CONTROL CONTENT TRANSMISSION METHOD AND NON-STORAGE TYPE
BROADCASTING SYSTEM

5

TECHNICAL FIELD

The present invention relates to a non-storage type information service system which automatically provides the user in real-time with a presenter required by the user to view desired content with a viewer, regardless of the presence or absence of the user's request, and more specifically to a non-storage type broadcasting system which automatically updates a viewer.

BACKGROUND ART

15 In an information service system, which provides users with a content viewing service, a browser is required for supplying content data to be provided to a presenter which is an input/output device allowing the user to access it through a physical phenomenon. This browser is required to be appropriately updated, as necessary, 20 in accordance with factors such as content data to be presented, a transmission path through which the content data is provided to the user, and the user's browser.

Examples of a conventionally implemented information service system include push-type information delivery services 25 on the Internet, such as PointCast(R) PointCast Network(R) and

Microsoft(R) Active Desktop(R), which involve the use of computers.

The push-type information delivery service can be classified into a storage type and a non-storage type. In the push-type information delivery service of the storage type, content, such as news or a weather forecast, which is broadcast by a broadcast station on the Internet, is received and stored by a reception device on the user side, e.g., a personal computer. In the push-type information delivery service of the non-storage type, content, such as news or a weather forecast, which is broadcast by a broadcast station on the Internet, is received by a reception device on the user side, e.g., a personal computer, and presented to the user in real-time. The user operates the reception device so as to activate the browser, which is a user interface for seeing the content, and thus can view the content stored in the receiver regardless of the storage type or the non-storage type.

Since different content processing methods are used for different services, the browser varies from one service to another. Therefore, it is possible to realize a user interface which is flexibly compatible with each service. If the browser is required to be upgraded, the user uses a function programmed to the browser itself to receive a new browser. Then, the current browser can be updated to the latest version by replacing it with the newly received browser.

Referring to FIGs. 24, 25, 26, 27, 28, 29, and 30, an exemplary conventional broadcasting system is described. As shown in a block

diagram of FIG. 24, a broadcasting system 2500 includes a transmission device 2510, a transmission path 120, and a non-storage type reception device 2520. The transmission device 2510 includes browser storages 2511, browser senders 2513, content storages 1113, content senders 2514, a multiplexer 115, and a sender 116.

A plurality of each of the browser storages 2511, the browser senders 2513, the content storage 1113, and the content senders 2514 are provided, and they are distinguished from each other by assigning a suffix (a lower-case alphabetic character) to their respective symbols. In the following descriptions, if a plurality of elements of the same type are present, they are distinguished from each other similarly by assigning a suffix to their respective symbols. Note that if there is no need to individually distinguish them, they are referred to by their respective generic names without assigning a suffix to their symbols.

In the example shown in FIG. 24, there are provided three of each of the browser storages 2511, i.e., 2511a, 2511b, and 2511c, the browser senders 2513, i.e., 2513a, 2513b, and 2513c, the content storages 1113, i.e., 1113a, 1113b, and 1113c, and the content senders 2514, i.e., 2514a, 2514b, and 2514c. Note that if there is no need to individually distinguish them, they are referred to by their respective generic names, such as the browser storages 2511, the browser senders 2513, the content storages 1113, and the content senders 2514.

FIGs. 25, 26, 27, and 28 show how data is stored in each element of the above broadcasting system 2500. The browser storage 2511 individually exists for each service. Three individual browser storages 2511a, 2511b, and 2511c are respectively provided for three services (S1, S2, and S3). The browser senders 2513 store browsers B corresponding to services (S) to be provided to the user, and send the browsers B to the multiplexer 115 according to a predetermined schedule.

The browser sender 2513a corresponding to the service S1 has a browser B(S1) stored therein. The browser B is a computer program written in native code (machine language) of a CPU (central processing unit) of the non-storage type reception device 2520. Similarly, the browser sender 2513b corresponding to the service S2 has a browser B(S2) stored therein, and the browser sender 2513c corresponding to the service S3 has a browser B(S3) stored therein. Note that the generic name "browser B" is used unless there is no need to distinguish their details.

A specific method for sending a browser, such as a transmission protocol or a transmission schedule, is uniquely defined for each service. Accordingly, the details of processing by the browser senders 2513 vary from one service to another. Therefore, the browser senders 2513a, 2513b, ... 2513n (where n is an arbitrary natural number) individually exist for their respective services.

In the example shown in FIG. 24, three individual browser

senders 2513a, 2513b, and 2513c correspond to three services S1, S2, and S3. The content storage 1113 stores content C belonging to a corresponding service. The content storage 1113 individually exists for each service. Three individual content storages 113a, 113b, and 113c are respectively provided for three services.

In the example shown in FIG. 25, the content storage 1113a corresponding to the service S1 has stored therein two service contents $C(S1,1)$ and $C(S1,2)$. The content storage 1113b corresponding to the service S2 has no content C stored therein. The content storage 1113c corresponding to the service S3 has stored therein service contents $C(S3,1)$, $C(S3,2)$, and $C(S3,3)$. Note that the generic name "service content C" is used as above unless there is no need to distinguish their details.

Referring back to FIG. 24, the content sender 2514 uses a predetermined method to send to the multiplexer 115 content $C(Sm,O)$ stored in the content storage 1113 associated with a corresponding service. Note that in content $C(Sm,O)$, Sm is a combination of symbol S representing a service, and a suffix m (an arbitrary natural number) assigned thereto for distinguishing individual services, while O (an arbitrary natural number) indicates the position in a sequence of a plurality of contents C belonging to service Sm.

In the sending device 2510 used in the conventional broadcasting system 2500, a method for sending a browser or content varies from one service to another. Accordingly, it is necessary to provide a plurality of individual plurality of browser senders

2513 and a plurality of individual content senders 2514 for their respective services. Further, browser B and content C are different in how they are sent. Accordingly, also for browser B and content C, it is necessary to provide an individual browser
5 sender 2513 and an individual content sender 2514.

The multiplexer 115 multiplexes and modulates a browser B outputted from the browser sender 2513 and service content C(Sm,O) outputted from the content storage 113 together into a form suitable for sending, and outputs a digital bit stream. The multiplexer
10 115 may consist of a multiplexing device and a modulation device in a digital broadcast station system.

The sender 116 sends a result of processing by the multiplexer 115 to the transmission path 120. A modem can be used as the sender 116 if the transmission path is a wired communication line.
15 Alternatively, if the transmission path is a broadcasting communications satellite and outer space, a parabolic antenna for sending may be used as the sender 116.

Next, the transmission path 120 is described. The transmission path 120 is a means for transmitting, to the
20 non-storage type reception device 2520, information such as content, a browser, etc., which is sent from the sending device 2510. For example, the transmission path may be an optical fiber, a cable of any type, a broadcasting communications satellite and outer space, or a package medium such as a DVD and a distribution channel
25 thereof.

Next, the non-storage type broadcasting system is described in detail. The non-storage type reception device 2520 includes a receiver 131, a demodulator/separator 132, a storage 133, a renderer 134, a presenter 135, an input device 136, a CPU 137,
5 and a memory 138.

The receiver 131 receives information such as content C, browser B, etc., which is sent to transmission path 120, and outputs a digital stream. The receiver 131 may be a modem or a combination of an antenna and a module for an ordinary digital broadcast
10 receiver.

The demodulator/separator 132 demodulates the digital bit stream outputted from the receiver 131, and separates multiplexed information. The demodulator/separator 132 carries out a process opposite to a process carried out by the multiplexer 115 of the
15 sending device 2510. The outputs of the demodulator/separator 132 are supplied to the storage 133 and the renderer 134, and can be read by the CPU 137.

Among the browser and the content outputted from demodulator/separator 132, the storage 133 stores only the browser.
20 This is because the non-storage type reception device 2520 displays, in real-time, the content C sent from the sending device 2510 and thus are not required to store the content C, while the browser B for presenting delivered content C has to always be prepared. The storage 133 is composed of, for example, a randomly accessible
25 storage medium, such as a hard disk. The contents of data stored

in the storage 133 can be read and changed by the CPU 137.

As shown in FIG. 26, the storage 133 has stored therein a total of three browsers B(S1), B(S2), and B(S3) which are supplied from the demodulator/separator 132. Note that the demodulator/separator 132 may output only the browser B to the storage 133, while the demodulator/separator 132 may output only the content C to the renderer 134.

In response to a request from the CPU 137, the renderer 134 renders OSD (on-screen display) and content, which is inputted from the demodulator/separator 132, into graphics that are to be displayed on the screen.

The presenter 135 presents graphics and/or audio, which are outputted from the renderer 134, in forms that are viewable/audible to the user. The presenter may be a CRT display, for example.

The input device 136 is a means operated by the user to provide the user's instruction to the non-storage type reception device 2520. As the input device 136, a combination of a remote control and a light reception portion of the remote control, a keyboard, a mouse, or the like can be used.

The CPU 137 is a central processing unit which is interconnected to each element of the non-storage type reception device 2520, and operable to execute a computer program stored in the memory 138 to control the entire non-storage type reception device 2520.

The memory 138 is composed of a RAM/ROM which is a

rewritable/non-rewritable semiconductor memory, and used as a region for storing data of processing by the CPU 137 or a computer program or data that is to be executed by the CPU 137. As illustrated in FIG. 27, the memory 138 has stored therein a browser list 2700 and an execution form 2651 which is native code (machine language) of the CPU 137.

Next, referring to FIG. 28, the browser list 2700 is described. The browser list 2700 is information in the form of a table which has a row for each service and having a column 2710 for browser file names and a column 2720 for service names. The browser list 2700 is used to make it possible to identify a browser used for utilizing a desired service from among information outputted from the demodulator/separator 132.

Next, referring to a flowchart shown in FIG. 29, a principle operation of the non-storage type reception device 2520 is described in detail.

At step S2801, service names in the browser list 2700 stored in the memory 138 are listed on the screen. The display onto the screen is carried out by the renderer 134.

At step S2802, it is detected based on the user's operation of the input device 136 that a service has been selected from the list displayed at step S2801.

At step S2803, for the unselected service detected at step S2802, the file name column of the browser list 2700 is referenced for browser B to identify the file name B(Sm) of the browser B

in the storage 133.

At step S2804, the file B(Sm) identified at step S2803 is executed. The browser is written in the native code of the CPU 137, and therefore can be directly executed by the CPU 137.

5 Next, referring to a flowchart shown in FIG. 30, a process performed by the non-storage type reception device 2520 for updating a browser is described.

At step S2901, browser B is activated by the CPU 137.

At step S2902, the receiver 131 and the demodulator/separators 10 132 receive browser B that is currently being sent out from the sending device 2510, and check the version thereof.

At step S2903, if it is determined that the browser B received and checked at step S2902 is of a newer version than the browser B that is currently being executed, the procedure proceeds to the next step S2904, and if not, then the procedure is terminated. 15

At step S2904, the non-storage type reception device 2520 receives the browser B that is being sent out, and the storage 133 temporarily stores the received browser B as a file as illustrated in FIG. 26.

20 At step S2905, the browser B that is being executed is replaced with the file that is being temporarily stored. Then, the browser B is reactivated to start executing the received new version of browser.

As a specific method for transmitting a browser which is 25 a computer native code, the above-illustrated conventional

broadcasting system uses a unique method which is different for each service. Accordingly, both the sending and reception devices are required to implement an individual method which is different for each browser which varies from one service to another.

5 That is, in the sending device, it is necessary to provide a different browser sender for each service. Also, in the reception device, the code of a browser of each service itself is typically programmed with a function of receiving a new browser for making a new service available and replacing the browser itself with the
10 received new browser.

As such, in order to realize the same transmission function as that of a browser which essentially carries out the same function for any service, slightly different browsers are required to be implemented.

15 Accordingly, various wastes arise with an increase in number of services. Specifically, the reception device is required to retain a plurality of similar program codes, and therefore storage capacities are wasted by the number of redundant browsers. Also, the reception device simultaneously activates similar processes,
20 so that computer resources cannot be efficiently utilized.

Also, the method for transmitting a browser varies from one service to another, and the implementation thereof is separately programmed for each browser for each service. It is troublesome that the user is required to previously obtain a browser for a
25 desired service manually by manually activating a file transmission

protocol such as FTP.

Also, the method for transmitting a browser differs from the method for transmitting content. Therefore, in the case of subscribing to a plurality of services, transmission of a browser
5 for one service might prevent transmission of a browser or content for another service.

Also, in the sending device, in order to realize browser transmission which is a function that is essentially consistent for any service, slightly different browser senders are required
10 for the number of services. Accordingly, the complexity of the sending device increases linearly with an increase of the number of services. This complexity leads to an increase in development cost and the complexity of management.

Also, there is no association between a plurality of browser
15 senders, therefore there is a possibility that a plurality of browsers might be sent out at the same time more than the transmission path can handle, overflowing the transmission path. Since a content sender individually exists for each service, content transmission might have a problem similar to the problem with the
20 browser transmission.

An object of the present invention is to provide a method and system for use in storage-type broadcasting in which a service consisting of a plurality of contents, which are transmitted via digital broadcasting or a computer network, is provided in
25 real-time for viewing, the method and system providing a user

interface which implements flexible services while protecting the safety of the services from invalid content.

DISCLOSURE OF THE INVENTION

5 To achieve the above object, the present invention has the following aspects.

 A first aspect of the present invention is directed to a non-storage type broadcasting system for providing one or more services composed of a content in real-time for viewing by a user
10 and providing a user interface unique to each of the services, the system comprising:

 a transmitter for sending out a control content, which implements the user interface, as a part or whole of the content; and

15 a receiver for receiving the sent control content and activating the received control content to execute the user interface, the user interface being transmitted/received as a content.

 In a second aspect based on the first aspect of the present
20 aspect, the control content is a browser for the content.

 In a third aspect based on the first aspect of the present aspect, the transmitter includes:

 a content sender for sending out a content containing the control content; and

25 a service attribute information sender for sending out

service attribute information indicating details of the services,
and

the receiver includes a control content identifier for
identifying the control content from among received contents based
5 on the received contents and the service attribute information.

In a fourth aspect based on the third aspect of the present
aspect, the content sending means includes a content header adder
for adding, to the content, a content header which defines details
of the content, and the reception means identifies the control
10 content from among the received contents based on rcontent headers
of the received contents.

In a fifth aspect based on the third aspect of the present
aspect, the transmission means further includes an electronic
signature device for applying an electronic signature to the
15 control content, the service attribute information sender sends
out a public key of the electronic signature in the service attribute
information, and

the receiver further includes a signature authentication
device for authenticating the electronic signature with the public
20 key contained in received service attribute information, the
control content being identified by authenticating the electronic
signature.

In a sixth aspect based on the fifth aspect of the present
aspect, the authentication by the electronic signature is performed
25 using a key independent of each service.

In a seventh aspect based on the third aspect of the present aspect, the content sender further includes a content ID space manager for sending out information which defines a partial space of an ID space of the content, and

5 the receiver further includes an identifier for identifying the control content based on whether a content ID falls within the partial space.

An eighth aspect of the present aspect is directed to a control content transmission method for use in a non-storage type
10 broadcasting system for providing one or more services composed of a content in real-time for viewing by a user, the method transmitting a user interface unique to each of the services, the method comprising:

 a step of sending out a control content, which implements
15 the user interface, as a part or whole of the content; and

 a step of receiving the sent control content and activating the received control content to execute the user interface.

In a ninth aspect based on the eighth aspect of the present aspect, the control content is a browser for the content.

20 In a tenth aspect based on the eighth aspect of the present aspect, the transmission step further includes: a content sending step of sending out a content containing the control content); and

 a service attribute information sending step of sending out
25 service attribute information indicating details of the services,

and

the reception step includes a control content step of identifying the control content from among received contents based on the received contents and the service attribute information.

5 In an eleventh aspect based on the tenth aspect of the present aspect, the content sending step includes a content header addition step of adding, to the content, a content header which defines details of the content, and

the reception step further includes a step of identifying
10 the control content from among the received contents based on content headers of the received contents.

In a twelfth aspect based on the tenth aspect of the present aspect, the transmission step further includes an electronic signature step of applying an electronic signature to the control
15 content,

the service attribute information sending step further includes a service attribute information sending step of sending out a public key of the electronic signature in the service attribute information, and

20 the reception step further includes:

a signature authentication step of authenticating the electronic signature with the public key contained in received service attribute information; and

a step of identifying the control content by
25 authenticating the electronic signature.

In a thirteenth aspect based on the twelfth aspect of the present aspect, the authentication by the electronic signature is performed using a key independent of each service.

In a fourteenth aspect based on the tenth aspect of the present aspect, the content sending step further includes a content ID space management step of sending out information which defines a partial space of an ID space of the content, and

the reception step further includes a step of identifying the control content based on whether a content ID falls within the partial space.

A fifteenth aspect is directed to a reception device for use in a non-storage type broadcasting system for providing one or more services composed of a content in real-time for viewing by a user and providing a user interface unique to each of the services, the reception device receiving a control content which implements the user interface, the control content being transmitted as a part or whole of the content and having added thereto service attribute information indicating details of the services, the reception device comprising:

a receiver for receiving contents transmitted from a transmitter;

an extractor for demodulating the received contents and extracting the contents and the service attribute information; and

a control content identifier for identifying the control

content from among the received contents based on the extracted contents and the service attribute information.

In a sixteenth aspect based on the fourteenth aspect, in the non-storage type broadcasting system, the content is
5 transmitted after being further added with a content header which defines details of the content,

the extractor further extracts the content header from the received contents, and

the control content identifier identifies the control
10 content from among the received contents based on the extracted content header.

In a seventeenth aspect based on the fifteenth aspect, in the non-storage type broadcasting system, the content the control content is signed with an electronic signature, and a public key
15 of the electronic signature is sent out in the service attribute information,

the reception device further includes a signature authentication device for authenticating the electronic signature with the public key contained in received service attribute
20 information, and

the control content identifier identifies the control content by authenticating the electronic signature.

In an eighteenth aspect based on the seventeenth aspect, the authentication by the electronic signature is performed using
25 a key independent of each service.

In a nineteenth aspect based on the fifteenth aspect, in the non-storage type broadcasting system, the content is added with information which defines a partial space of an ID space of the content, and

5 the identifier further identifies the control content based on whether a content ID falls within the partial space.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram showing a structure of a non-storage
10 type broadcasting system according to a first embodiment of the present invention.

FIG. 2 is a diagram for explaining service attribute information stored in a service attribute storage shown in FIG. 1.

15 FIG. 3 is a schematic diagram showing examples of content bodies stored in a content storage shown in FIG. 1.

FIG. 4 is a table for explaining service attribute information in the non-storage type broadcasting system shown in FIG. 1.

20 FIG. 5 is a diagram for explaining a service content in the non-storage type broadcasting system shown in FIG. 1.

FIG. 6 is a diagram for explaining a browser content in the non-storage type broadcasting system shown in FIG. 1.

25 FIG. 7 is a schematic diagram showing examples of contents stored in a storage shown in FIG. 1.

FIG. 8 is a schematic diagram showing information stored in a memory shown in FIG. 1.

FIG. 9 is a flowchart showing an operation of executing a browser content by a non-storage type reception device shown in
5 FIG. 1.

FIG. 10 is a flowchart showing an operation of updating a browser content by the non-storage type reception device shown in FIG. 1.

FIG. 11 is a block diagram showing a structure of a non-storage
10 type broadcasting system according to a second embodiment of the present invention.

FIG. 12 is a table for explaining service attribute information in the non-storage type broadcasting system shown in FIG. 11.

15 FIG. 13 is a diagram for explaining a browser content in the non-storage type broadcasting system shown in FIG. 11.

FIG. 14 is a flowchart showing an operation of updating a browser content by a non-storage type reception device shown in FIG. 11.

20 FIG. 15 is a block diagram showing a structure of a non-storage type broadcasting system according to a third embodiment of the present invention.

FIG. 16 is a table for explaining service attribute information in the non-storage type broadcasting system shown in
25 FIG. 15.

FIG. 17 is a diagram for explaining a browser content in the non-storage type broadcasting system shown in FIG. 15.

FIG. 18 is a flowchart showing an operation of updating a browser content by a non-storage type reception device shown in
5 FIG. 15.

FIG. 19 is a block diagram showing a structure of a non-storage type broadcasting system according to a fourth embodiment of the present invention.

FIG. 20 is a table for explaining service attribute
10 information in the non-storage type broadcasting system shown in FIG. 19.

FIG. 21 is a table for explaining a content ID in the non-storage type broadcasting system shown in FIG. 19.

FIG. 22 is a diagram for explaining a browser content in
15 the non-storage type broadcasting system shown in FIG. 19.

FIG. 23 is a flowchart showing an operation of updating a browser content by a non-storage type reception device shown in FIG. 19.

FIG. 24 is a block diagram showing a structure of a
20 conventional non-storage type broadcasting system.

FIG. 25 is a schematic diagram showing examples of contents stored in content and browser storages shown in FIG. 24.

FIG. 26 is a schematic diagram showing an example of a content and browsers stored in a non-storage shown in FIG. 24.

25 FIG. 27 is a schematic diagram showing information stored

in a memory shown in FIG. 24.

FIG. 28 is a table for explaining service attribute information in the non-storage type broadcasting system shown in FIG. 24.

5 FIG. 29 is a flowchart showing an operation of executing a browser by a reception device shown in FIG. 24.

FIG. 30 is a flowchart showing an operation of updating a browser by the reception device shown in FIG. 24.

10 BEST MODE FOR CARRYING OUT THE INVENTION
(First Embodiment)

Referring to FIGs. 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10, a non-storage type broadcasting system according to a first embodiment of the present invention is described.

15 As shown in FIG. 1, a non-storage type broadcasting system 100 according to the first embodiment includes a transmission device 110, a transmission path 120, and a non-storage type reception device 130. As will be described later, the non-storage type reception device 130 stores a browser but does not store content
20 itself, and therefore it is referred to as the non-storage type.

The transmission device 110 includes a service attribute storage 111, a content header adder 112, content storages 113, a content sender 114, a multiplexer 115, and a sender 116. A plurality of content storages 113 are provided, and they are
25 distinguished from each other by assigning a suffix (a lower-case

alphabetic character) to their respective symbols. In the following descriptions, if a plurality of elements of the same type are present, they are distinguished from each other similarly by assigning a suffix to their respective symbols. Note that if
5 there is no need to individually distinguish them, they are collectively referred to as the content storages 113, for example.

As described above, if there are a plurality of elements of the same type, they are distinguished herein similarly by adding a suffix to their respective symbols. Further, if there is no
10 need to individually distinguish them, they are referred to by their respective generic names without assigning a suffix to their symbols

Each of the content storages 113 has stored therein a content body Dc which is substantive data of a service. Further, the
15 content storages 113 sequentially supply the content bodies Dc to the content sender 114.

The content sender 114 sends the supplied content bodies Dc to the content header adder 112.

The content header adder 112 adds to the supplied content
20 bodies Dc, a content header Hc consisting of management information of the content bodies Dc including header information, thereby generating content C. The generated content C is supplied to the multiplexer 115.

The multiplexer 115 multiplexes service attribute
25 information Isp supplied from the service attribute storage 111

with content C outputted from the content header adder 112, and supplies it to the sender 116.

The sender 116 modulates the service attribute information Isp modulated with the content C into a digital bit stream in the form adapted to the transmission path, and then transmits it to the transmission path 120. The sender 116 may be a modem if the transmission path is a wired communication line, or may be a parabolic antenna for sending if the transmission path is a broadcasting communications satellite and outer space.

The transmission path 120 is a means for transmitting to the non-storage type reception device 130 information, such as the content C or the browser B, which has been transmitted from the transmission device 110. For example, the transmission path can be composed of an optical fiber, a cable of any type, a broadcasting communications satellite and outer space, or a package medium such as a DVD and a distribution channel thereof.

The non-storage type reception device 130 includes a receiver 131, a demodulator/separator 132, a storage 133, a renderer 134, a presenter 135, an input device 136, a CPU 137, and a memory 138. Note that the storage 133, the renderer 134, the input device 136, the CPU 137, and the memory 138 are interrelated with each other via a data buss.

The receiver 131 receives information, such as the content C or the browser B, which has been transmitted to the transmission path 120, and outputs a digital stream. The receiver 131 can be

composed of a mobile communication module, such as a communication modem, a cable modem, or Bluetooth, and an antenna and tuner module for an ordinary digital broadcast receiver.

The demodulator/separator 132 demodulates the digital bit stream outputted by the receiver 131, and separates multiplexed information into pieces of information. The demodulator/separator 132 performs a process opposite to that of the multiplexer 115 of the transmission device 110. The outputs of the demodulator/separator 132 are supplied to the storage 133 and the renderer 134, and can be read by the CPU 137.

The storage 133 is, for example, a randomly accessible storage medium such as a hard disk, and information thereon can be read and modified by the CPU 137. Among service content $C(Sm, O)$ and browser content $C(Sm, Bflg)$ outputted from the demodulator/separator 132, the storage 133 stores only the browser content $C(Sm, Bflg)$. This is because the non-storage type reception device 130 displays, in real-time, the service content $C(Sm, O)$ transmitted from the transmission device 110 and thus are not required to store the service content $C(Sm, O)$, while the browser content $C(Sm, Bflg)$ containing the browser B for presenting the distributed service content $C(Sm, O)$ has to always be prepared.

The service content $C(Sm, O)$ corresponds to an essence of a service provided by the non-storage type broadcasting system 100, e.g., a program or data which the user enjoys viewing. The browser content $C(Sm, Bflg)$ implements a user interface for

allowing the user to actually enjoy a provided service with the non-storage type reception device 130. The browser content C (Sm,Bflg) is a control code executable under a program code execution environment provided by the non-storage type reception
5 device 130, and is intended to process the service content C (Sm,O).

In response to a request from the CPU 137, the renderer 134 renders OSD (on-screen display) and the service content C (Sm,O), which is inputted from the demodulator/separator 132, into graphics that are to be displayed on the screen.

10 The presenter 135 presents outputs of the renderer 134 in forms that are viewable/audible to the user. The presenter may be a CRT display, for example.

The input device 136 is used by the user to provide an instruction for operation to the non-storage type reception device
15 130. The input device 136 can be composed of a combination of a remote control and a light reception portion of the remote control, a keyboard, a mouse, or the like.

The CPU 137 is a central processing unit which is interconnected to each element of the non-storage type reception
20 device 130, and operable to execute a computer program stored in the memory 138 to control the entire non-storage type reception device 130.

The memory 138 is composed of a RAM/ROM which is a rewritable/non-rewritable semiconductor memory, and used as a
25 region for storing data of processing by the CPU 137 or a computer

program or data that is to be executed by the CPU 137.

Referring to FIGs. 2, 3, 4, and 5, data stored in each element of the above-described non-storage type broadcasting system 100 is described.

5 As shown in FIG. 2, the service attribute storage 111 has stored therein service attribute information Isp indicating an attribute of each service provided by the non-storage type broadcasting system 100.

10 As shown in FIG. 3, the content storage 113 has stored therein a content body Dc which is an essence of the content C which is data constituting a corresponding service. The content storage 113 individually exists for each service. Specifically, there are provided three individual content storages 113a, 113b, and 113c corresponding to three different services.

15 The content storage 113a corresponding to service S1 has stored therein two content bodies Dc(S1,1) and Dc(S1,2) and one browser content body Dc(S1,Bflg). Note that an arbitrary content body Dc is represented as a content body Dc(Sm,O) by assigning suffix m (an arbitrary natural number) to symbol S, which denotes
20 a service, to identify an individual service, and assigning parameter O (an arbitrary natural number), which indicates the position in a sequence of a plurality of contents C constituting service Sm.

25 The browser content body Dc(S1,Bflg) is obtained by generating browser B(S1), which is used for service S1, in the

same format as that of the content body Dc, and can be represented by the body Dc(Sm,Bflg) in a manner similar to the service content body Dc(Sm,O).

5 The browser content body Dc(Sm,Bflg) is different in details from the service content body Dc(Sm,O). Specifically, it contains, instead of O, parameter Bflg which indicates service attribute information Isp. Note that the service attribute information Isp will be described in detail later with reference to FIG. 4.

10 However, the browser content body Dc(Sm,Bflg) is in the same data format as that of the service content body Dc(Sm,O), therefore the content storage 113 cannot distinguish them and handles both of them as the content body Dc. In this context, both the browser content body Dc(Sm,Bflg) and the service content body Dc(Sm,O) are collectively referred to as content body Dc. For
15 simplification, the service content body Dc(Sm,O) and the browser content body Dc(Sm,Bflg) are referred to as service content body DcS and browser content body DcB, respectively, if need arises. Further, the browser content C(Sm,Bflg) containing the browser content body Dc(Sm,Bflg) is simply referred to as browser content
20 Cb, and the service content C(Sm,O) containing the service content body Dc(Sm,O) is simply referred to as service content Cs.

Next, referring to FIGs. 4, 5 and 6, service content Cs and browser content Cb, which are outputted from the content header adder 112, are described in detail. FIG. 4 illustrates by way
25 of example service attribute information Isp stored in the service

attribute storage 111. The service attribute information Isp is information related to all services (S1, S2, ... , Sm) provided by the non-storage type broadcasting system 100. In the table, each row corresponds to a service, and contains a service ID and
5 a service name.

For example, the top row shows a service having the service ID "S1" and the service name "TODAY'S NEWS". Similarly, the second row shows a service having the service ID "S2" and the service name "RECOMMENDATION INFORMATION", and the third row shows a
10 service having the service ID "S3" and the service name "NEW CAR GUIDE". The service ID is represented by a unique value in the service attribute information Isp, and can be used for identifying the service.

FIG. 5 schematically shows a data structure of the service
15 content Cs, and FIG. 6 schematically shows a data structure of the browser content Cb. The content C is composed of a content body Dc, which is essential data of the content C it self, and a content header Hc, which is metadata associated with the content body Dc.

20 In FIG. 5, a service content body DcS of content C(S1,1), which is the service content Cs, is associated with a content header HcS. In FIG. 6, a browser content body DcBn of C(S1,Bflg), which is the browser content Cb, is associated with a content header HcB.

25 The content headers Hc (i.e., HcS and HcB) are represented

in the form of a table in which a row consists of a set of an item and a value. The content header HcS of the service content Cs (i.e., C(S1,1)) includes four rows, i.e., SERVICE ID, CONTENT ID, CONTENT VERSION, and CONTENT NAME.

5 The service ID corresponds to a service ID in the service attribute information Isp illustrated in FIG. 4, and identifies to which service S the content C corresponds. In present example, the service ID is S1, and therefore it is indicated that the content C is the "TODAY'S NEWS" service shown in FIG. 4.

10 The content ID is used for identifying the content C itself. Accordingly, the content ID is managed, for example, so as to avoid overlaps on the sending side, and has a unique value at least in the non-storage type broadcasting system 100.

 The content version indicates whether the content C is old
15 or new. The content version has an individual value for each content ID. If it is desired to replace a current content Cc, which is currently stored in the storage 133 that is to be described in detail later with reference to FIG. 7, with a new content Cn to which it is desired to be updated, a content version having
20 a value greater than that of the current content Cc is assigned to the new content Cn when the content C is sent out.

 The content name is an item to be read as a character string when the user selects the content. In the present example, it is indicated that the content body Dc is the "WEATHER FORECAST".

25 The above-described service ID, content ID, content version,

and content name are commonly contained in the service content header HcS and the browser content header HcB.

However, as shown in FIG. 6, the browser content header HcB of the browser content Cb further contains a browser content flag
5 row.

The browser content flag Bflg is an item for indicating that the content C is the browser content Cb, and corresponds to the above-described parameter Bflg. The browser content flag of the browser content header HcB is indicated as "True", therefore it
10 is indicated that the content body Dc is the browser content body DcB.

In the browser content Cb illustrated in FIG. 6, the service ID is S1 and the content ID is 101. Therefore, it is indicated that the browser content Cb for the service "TODAY'S NEWS" contains
15 the content body Dc having the content ID "101". Thus, three of the service ID, the content ID, and the browser content flag are information at least required for defining that the content C is the browser content Cb.

Hereinbelow, processes which the service attribute storage
20 111, the content header adder 112, the content storage 113, and the content sender 114 perform based on the above-described data structure of the content C are described.

The content sender 114 sends to the content header adder 112 content bodies Dc sequentially supplied by the content storage
25 113. The content header adder 112 assigns a content header Hc

(HcS, HcB) to a content body Dc (i.e., DcS, DcB) to generate content C (i.e., Cs, Cb), and supplies it to the multiplexer 115.

The following description is given by referring to FIGs. 7 and 8 with respect to what type of data is stored in each element of the non-storage type reception device 130 when content C (service content Cs, browser content Cb) is transmitted from the transmission device 110 via the transmission path 120.

A digital bit stream transmitted from the transmission device 110 via the transmission path 120 and containing the content C and the service attribute information Isp is first received by the receiver 131 of the non-storage type reception device 130, and then supplied to the demodulator/separator 132.

The demodulator/separator 132 demodulates the digital bit stream, and separates and reproduces the content C and the service attribute information Isp which are contained in the demodulated digital bit stream. Further, the demodulator/separator 132 extracts a content header Hc from the reproduced content C, and outputs it to a bus together with the service attribute information Isp.

The reproduced content C is temporarily retained in the demodulator/separator 132. Note that where it is necessary to distinguish the reproduced content C from content C having been previously stored in the storage 133, the content C retained in the demodulator/separator 132 is referred to as "demodulated content Cn" and the content C stored in the storage 133 is referred

to as "stored content Ca".

The content C reproduced by the demodulator/separator 132 is supplied at a predetermined time and stored into the storage 133. As shown in FIG. 7, in the storage 133, among contents C (i.e., Cs, Cb) of services outputted from the content header adder 112 to the multiplexer 115, only the browser content Cb is stored in the storage 133. Specifically, as described with reference to FIG. 7, only the browser content Cb is stored as the stored content Ca stored in the storage 133.

That is, among contents C(S1,Bflg), C(S1,1), C(S1,2), C(S2,Bflg), C(S3,Bflg), C(S3,1), C(S3,2), and C(S3,3) which respectively correspond to a total number of eight content bodies Dc(S1,Bflg), Dc(S1,1), Dc(S1,2), Dc(S2,Bflg), Dc(S3,Bflg), Dc(S3,1), Dc(S3,2), and Dc(S3,3), which are stored in the content storages 113a, 113b, and 113c shown in FIG. 3, the contents C(S1,Bflg), C(S2,Bflg), and C(S3,Bflg) are stored in the storage 133.

Further, the storage 133 is controlled by the CPU 137 so as to output the browser content body Dc(S,Bflg), which is contained in the stored browser content Cb, to a bus or the renderer 134.

The renderer 134 generates, based on the supplied information, a video signal for rendering an image on the presenter 135.

As shown in FIG. 8, the memory 138 has stored therein the service attribute information Isp, a browser content execution interpreter 271, and an execution form 2651 which is a native code

of the CPU 137. The browser content execution interpreter 271 is used for executing content as a computer high-level language program.

Next, referring to a flowchart shown in FIG. 9, an operation of activating the browser content C_b contained in the content C received by the non-storage type reception device 130 is described.

At step S501, a list of service names of services described by the service attribute information Isp outputted from the demodulator/separator 132 is displayed on the screen of the presenter 135.

At step S502, a service selected from the list of service names displayed at step S501 is detected based on the user's operation of the input device 136. Specifically, the service selected by the user is identified based on the user's operation of the input device 136 and the list of service names.

At step S503, a service ID corresponding to the service identified at step S502 is obtained from the service attribute information Isp.

At step S504, all contents C stored in the storage 133 are searched through in order to select a content C_b having the service ID obtained at step S503 and a content header HcB with a browser content flag Bflg indicated as "True".

At step S505, the browser content C_b selected at step S504 is executed by the browser content execution interpreter 271.

In this manner, SINCE the browser content flag Bflg is

contained in the content header HcB of the browser content Cb, the browser content Cb can be distinguished from other ordinary service contents Cs. Also, by designating an individual service S, a browser content Cb corresponding thereto (i.e., C(Sm,Bflg))
5 can be activated as a browser.

Next, referring to a flowchart shown in FIG. 10, an operation performed by the non-storage type reception device 130 for updating the browser content Cb is described.

At step S601, the demodulator/seperator 132 reproduces the
10 content Cn (i.e., Cb, Cs), the service attribute information Isp, and the content header Hc (i.e., HcB, HcS).

At step S608, it is determined whether the browser content flag Bflg of the newly separated content Cn is "True". If it is "False", the procedure returns to the above step S601. Note that
15 if there is no browser content flag Bflg, i.e., in the case of the service content Cs, it is determined at this step as being false. If the browser content flag Bflg is true, i.e., in the case where the new content Cn is the browser content Cb, the procedure proceeds to the next step S602.

20 At step S602, the content header Hc separated at step S601 is outputted to a bus. Specifically, only the browser content header HcB of the browser content Cb is outputted.

At step S603, the storage 133 is searched through to select a new content Cc (browser content Cb) corresponding to the separated
25 content header Hc. Specifically, the current content Cc (browser

content Cb) having the same service ID as that written in the separated content header Hc is selected from among all browser contents Cb stored in the storage 133.

At step S604, if it is determined that the storage 133 includes
5 no browser content Cb having the same content ID, the procedure proceeds to the next step S605.

At step S605, a new browser content Cn (Cb) to which none in the storage 133 corresponds is stored anew as the current browser content Cc (Cb). Then, the procedure returns to the above step
10 S601.

On the other hand, at step S604, if it is determined based on a search result of step S602 that the storage 133 contains a browser content Cb (current content Cc) having the same content ID, the procedure proceeds to step S606.

At step S606, whether the new browser content Cn (Cb) separated at step S601 is newer than the current browser content Cn (Cb) stored in the storage 133 is determined based on a content version contained in the browser content header HcB outputted at step S602. If it is determined as being new, the procedure proceeds
15 to the next step S606.
20

At step S607, the current browser content Cn (Cb) stored in the storage 133 is updated by the new browser content Cn (Cb) retained in the demodulator/separator 132. Note that at this point, the new browser content Cn (Cb) becomes a current browser content
25 Cc. Then, the procedure returns to step S601.

On the other hand, at step S606, if it is determined that the newly demodulated browser content Cn (Cb) is not newer than the current new browser content Cc (Cb) stored in the storage 133, the above step S607 of updating the current browser content Cn (Cb) is skipped, and the procedure returns to step S601.

As described above, in the non-storage type broadcasting system 100, the transmission device 110 sends the browser content Cb, which provides a user interface to each service Sm, as part or whole of content C constituting the service Sm. Then, the non-storage type reception device 130 receives the sent content C, and activates the browser content Cb contained in the received content C, thereby executing the user interface. Note that the procedure shown in FIG. 10 may be executed as part of step S504 of the flowchart shown in FIG. 9, or may be executed only when it is necessary to update the browser content Cb.

Then, the browser content Cb is automatically updated or retained at a time point when the one of a version newer than that stored in the non-storage type reception device 130 is distributed, or when the one of any version other than those retained in the non-storage type reception device 130 is distributed. Specifically, the process of updating the content C is carried out only when the content C is the browser content Cb. This mechanism is not a function for each specific service, but is a function unique to the entire non-storage type reception device 130 and common to all services. Accordingly, by merely realizing

the function of updating the browser content Cb, it is possible not only to process service contents Cs for all the services in real-time but also to update and store the browser, i.e., the browser content Cb.

5 In other words, in the present embodiment, by transmitting the browser content Cb, which is a user interface for each service, as part of content, it is made possible to eliminate of the need of preparing a special mechanism for transmitting a user interface for each service and to update the user interface as if content
10 is updated.

(Second Embodiment)

Hereinbelow, referring to FIGs. 11, 12, 13, and 14, a storage type broadcasting system according to a second embodiment of the present invention is described. As shown in FIG. 11, similar to
15 the non-storage type broadcasting system 100 as shown in FIG. 1, a non-storage type broadcasting system 700 according to the present embodiment includes a transmission device 710, a transmission path 120, and a non-storage type reception device 730.

The transmission device 710 is such that the transmission
20 device 110 of the non-storage type broadcasting system 100 additionally includes an electronic signature device 711. The non-storage type reception device 730 is such that the non-storage type reception device 130 additionally includes a KP extractor 740, an Sg extractor 741, and a decoder 721. Hereinbelow, features
25 specific to the non-storage type broadcasting system 700 are mainly

described, and the descriptions of features common to the non-storage type broadcasting system 100 are omitted.

The digital signature device 711 manages electronic signature keys, and applies an electronic signature (digital
5 signature) to browser content Cb. As an electronic signature, a technique for public key cryptography is used. In this technique, a set of a secret key and a public key are generated for a distributor of information, and while only the public key is distributed to users of the information, the secret key is kept in secret. The
10 distributor uses the secret key to apply an electronic signature to information that is desired to be distributed (the information to which the electronic signature has been applied is provided in the form of a combination of original information and the electronic signature). The information to which the electronic
15 signature is applied and which is received by the user and the distributor's public key are used to make it possible to authenticate that the distributor having the secret key has applied the electronic signature to that information.

The electronic signature device 711 has a combination of
20 a secret key and a public key for each service. The electronic signature device 711 uses the secret key of the service to apply an electronic signature to the browser content Cb of the service. Specifically, the electronic signature device 711 is connected to the content storage 113 and outputs, in accordance with services
25 of the content bodies Dc respectively supplied from the content

storages 113a, 113b, and 113c, public keys KP different among the services to the service attribute storage 111.

Specifically, as shown in FIG. 13, by adding the item of ELECTRONIC SIGNATURE to the content header HcB of the browser content Cb, the electronic signature is contained in the browser content Cb. Accordingly, unlike the browser content Cb, browser content Cbc (S1,Bsgn) has a content header with an electronic signature row. A value column of the electronic signature row contains information (electronic signature) about a result of applying an electronic signature to a corresponding content body Dc with a secret key of service S1 to which the content belongs.

The content header HcB assigned with the electronic signature is specified as content header HcBc. Further, in order to distinguish a browser content with an electronic signature item from other browser contents Cb, that browser content is referred to as browser content Cbc, and represented by C(Sm,Bsgn).

The service attribute storage 111 generates service attribute information Ispc including a public key KP, and outputs it to the multiplexer 115. FIG. 12 shows the structure of the generated service attribute information Ispc. The service attribute information Ispc has a column for storing public keys KP. The public keys stored correspond to three services S1, S2, and S3.

Further, the electronic signature device 711 generates electronic signature Sg, and outputs it to the content header adder

112. The content header adder 112 generates a browser content Cbc with a signature based on the content body Dc supplied from the content sender 114 and the electronic signature Sg supplied from the electronic signature device 711, and outputs, to the multiplexer 115, content C consisting of the browser content Cbc with a signature and a service content Cs.

The non-storage type reception device 730 reproduces the content C (Cs and Cbc), content headers Hc (HcS,HcBc), and the service attribute information Ispc from a digital bit stream received by the demodulator/separator 132. The content C (Cs and Cbc) is retained in the demodulator/separator 132, the content headers Hc (HcS and HcBc) are outputted to a bus and the Sg extractor 741, and the service attribute information Ispc is outputted to the KP extractor 740.

15 The Sg extractor 741 extracts the electronic signature Sg from the content header Hc the browser content C (Sm,Bflg) among the supplied content headers Hc, and outputs it to the bus.

The KP extractor 740 extracts the public key KP from the service attribute information Ispc, and outputs it to the bus.

20 The decoder 721 authenticates an electronic signature in the content header HcBc based on the public key KP supplied via the bus.

As described above, the non-storage type reception device 730 is able to authenticate the correctness of the received browser content Cbc based on the electronic signature Sg of the browser

content Cbc and the public key KP in the service attribute information Ispc. Thus, a content creator of each service is able to ensure that the browser content Cbc is encrypted with a secret key unique to the service.

5 Next, an operation of the non-storage type reception device 730 according to the second embodiment is described. The operation of the non-storage type reception device 730 for activating the browser content Cb contained in the received content C is basically the same as that of the non-storage type reception device 130
10 according to the first embodiment described with reference to the flowchart shown described with reference to FIG. 9, except that the service attribute information Isp is changed to Ispc and the target content C is changed to the browser content Cbc. Accordingly, the description of the operation is omitted.

15 Now, referring to a flowchart shown in FIG. 14, an operation performed by the non-storage type reception device 730 for updating the browser content Cb is described. The operation performed by the non-storage type reception device 730 for updating the browser content Cb is similar to the operation performed by the non-storage
20 type reception device 130 for updating the browser content Cb as described with reference to FIG. 10. Specifically, in the flowchart shown in FIG. 14, steps S1001, S1002, S1003, and S1004 are additionally provided between steps S603 and S604 in the flowchart shown in FIG. 10.

25 That is, as described above, after the processes of steps

S601, S602, and S603, the browser content Cb is targeted for processing, and the procedure proceeds to the additionally provided step S1001.

At step S1001, an obtained browser content Cbc is referenced
5 for the row of the electronic signature stored in the content header HcB. If there is any electronic signature item, the procedure proceeds to step S1002.

At step S1002, it is authenticated whether the electronic signature Sg of the browser content Cbc has been applied with a
10 secret key corresponding to a public key KP of a corresponding service. The public key KP used therefor is the one obtained by selecting the row of that corresponding service from the service attribute information Ispc. The procedure proceeds to the next step S1003.

15 If it is determined at step S1003 based on a result of the authentication at step S1002 that the electronic signature Sg is not correct, the procedure proceeds to step S1004.

At step S1004, a warning that the browser content Cbc with a signature is invalid is displayed, and the procedure is terminated.
20 Accordingly, in any cases where it is determined that there is any abnormality in the electronic signature (No at step S1001 or S1003), the warning is issued at this step, and then the procedure proceeds to the above-described step S603.

In contrast, if it is determined at step S1003 that the
25 electronic signature is correct, the above-described step S1004

is skipped, and the procedure proceeds to steps S604, S605, S606, and S607, as in the case of the first embodiment.

As described above, in the non-storage type broadcasting system 700 according to the present embodiment, as in the non-storage type broadcasting system 100, a sent content C is received, and a user interface is executed by activating a browser content C_b contained in the received content C. However, the non-storage type broadcasting system 700 employs an electronic signature technique so that only the one in possession of the secret key of a target service is able to send a browser content C_b executable in the non-storage type reception device 730.

Thus, even if the one who is out of possession of the secret key for the target service attempts to store whatever content into the content storage, it is possible to prohibit intentionally transmitting an erroneous browser content C_b to execute it in the non-storage type reception device 730. Also, it is possible to prohibit a browser content C_b to which an electronic signature is applied with a secret key of a service other than the target service from being executed as the browser content C_b of the target service.

Note that although the electronic signature is applied only to the content body for the reason of simplifying the mechanism, the content header may be contained in the content body such that the electronic signature is applied thereto.

Note that a simple electronic signature method which does

not use public key encryption may be used. For example, a secret key may be generated for each service, and the secret key may be stored in the reception device by any method. Information which provides an indication of the browser content C_b may be encrypted with the secret key for each service, and the information may be decrypted on the reception device side to determine that it is the browser content C_b.

As described above, in the present embodiment, it is possible to ensure the reliability of content C distributed in the non-storage type broadcasting system 700 since the system includes: the electronic signature device 711, which applies an electronic signature to the browser content C_{bc}, which is content for providing a user interface for each service; the content storage 113 which sends the content body D_c containing browser content body D_{Cbc}; the service attribute storage 111 which sends the public key KP of the electronic signature S_g as the service attribute information I_{spc} common to all the services; and the non-storage type reception device 730 which authenticates the electronic signature S_g to determine that the content C is the browser content C_{bc}.

That is, in the present embodiment, the electronic signature is used for a mechanism of designating the browser content C_b from among a plurality of contents C belonging to one service, and a key used for the electronic signature is prepared for each service provider, so that no one but the provider can designate the browser content C_b for that service. As a result, it is possible to prevent

the browser content Cb from being incorrectly designated by the one other than the service provider who has malicious intent.

(Third Embodiment)

Next, referring to FIGs. 15, 16, 17, and 18, a storage type
5 broadcasting system according to a third embodiment of the present invention is described. As shown in FIG. 15, similar to the non-storage type broadcasting system 100 shown in FIG. 1, a non-storage type broadcasting system 1000 according to the present embodiment includes a transmission device 1010, a transmission
10 path 120, and a non-storage type reception device 130. The transmission device 1010 is such that the content header adder 112 of the transmission device 110 in the non-storage type broadcasting system 100 is replaced with a content header adder 112i. Hereinbelow, features specific to the non-storage type
15 broadcastingsystem1000aremainlydescribed, and the descriptions of features common to the non-storage type broadcasting system 100 are omitted.

In addition to the functions of the content header adder 112, the content header adder 112i has the following functions.
20 Specifically, for a plurality of content bodies Dc stored in the content storages 113a, 113b, and 113c, browser contents Cbi are designated, and content IDs and content versions of the designated contents (browser contents Cbi) are extracted as browser identification information Bid. Then, the content header adder
25 112i outputs the extracted browser identification information Bid

to the service attribute storage 111.

As shown in FIG. 16, the service attribute storage 111 writes the content IDs and content versions of the browser contents Cb into the service attribute information Isp based on the browser
5 identification information Bid. Specifically, the service attribute information Ispi has a browser content designation information column consisting of a column indicating the content IDs and a column indicating content versions.

The browser content designation information is designated
10 for each service, thereby making it possible to identify a browser content Cb(Cbi) among contents. In this manner, the service attribute information Isp in which the content ID and content version of the browser content Cbi are written is identified as the service attribute information Ispi.

15 The designation of the browser content Cbi is not performed based on the browser content flag of the content header HcBi, but it is performed by the content header adder 112i selecting the content.

FIG. 17 shows an exemplary content header HcBi. Since the
20 designation of the browser content Cbi is not performed based on the browser content flag Bflg, the content header HcBi does not have a row for the browser content flag Bflg. On the content header, it is not possible to distinguish between service content Cs and browser content Cb. In the present embodiment, although it seems
25 that the browser content Cb is recognizable based on the row of

the content name because a human being recognizes the meaning of a character string by reading it, it is not recognizable on the level of the transmission device 1010 and the non-storage type reception device 130.

5 Next, an operation of the non-storage type reception device 130 according to the present embodiment is described. The operation of the non-storage type reception device 730 for activating the browser content Cb contained in a received content C is basically the same as that of the non-storage type reception
10 device 130 according to the first embodiment described with reference to the flowchart shown described with reference to FIG. 9, except that the service attribute information Isp is changed to Ispc and the target content C is changed to the browser content Cbc. Accordingly, the description of the operation is omitted.

15 As described above, in the non-storage type broadcasting system 1000, the non-storage type reception device 130 is configured in the same manner as that of the non-storage type broadcasting system 100, but there are slight differences in operation because program codes and data that are to be stored
20 are different. Now, referring to a flowchart shown in FIG. 18, an operation of updating browser content Cb by the non-storage type reception device 130 according to the present embodiment is described. As shown in FIG. 18, in the operation of updating the browser content Cb in accordance with the present embodiment, step
25 S608 of the flowchart shown in FIG. 10 is replaced by steps S1401

and S1402.

Specifically, at step S1401, for a service selected by the user (S502, S503), the service attribute information Ispi is referenced to retrieve browser content designation information based on the content C (i.e., Cn) demodulated and the content header Hc separated at step S601.

At step S1402, the storage 133 is searched through to select content C (browser content Cbi) in possession of the content ID and content version of the browser content designation information retrieved at step S1401. Then, the selected content C is determined as the browser content Cbi, and thereafter the procedure proceeds to the above-described step S604. On the other hand, as a result of the search, if no corresponding content C (browser content Cbi) is selected, the procedure proceeds to step S601.

The designation of the browser content Cbi is performed by the content header adder 112i. Accordingly, by appropriately managing the content header adder 112i, it is possible to prevent the browser content Cbi from being repeatedly designated or to prevent a content C for a different service from being erroneously designated as the browser content Cbi.

As described above, in the present embodiment, it is possible to realize the non-storage type broadcasting system 100 which ensures that the browser content Cbi is distinguished from the service content Cs by providing the system with the content sender 114 for sending a content body constituting each service; the

service attribute storage 111 which sends the service attribute information Ispi containing, as browser content designation information common to all services, the browser identification information Bid for designating the browser content body DCb from a plurality of content bodies; and the non-storage type reception device 130 which determines whether content C is the browser content Cbi, based on the browser content designation information.

Specifically, in the present embodiment, the browser content Cbi is designated based on control content designation information common to a plurality of services, and the browser content designation information is provided independently of content and content headers. Thus, it is possible to prevent the browser content Cb from being incorrectly designated for a different service regardless of contents of the content storage. For example, if the operation is such that the browser content Cb designation information is managed under a contract broadcaster, rather than under each service provider, it is possible to prevent the browser content Cb from being incorrectly designated by a plurality of service providers.

(Fourth Embodiment)

Next, referring to FIGs. 19, 20, 21, 22, and 23, a storage type broadcasting system according to a fourth embodiment of the present invention is described. As shown in FIG. 19, similar to the non-storage type broadcasting system 100 shown in FIG. 1, a non-storage type broadcasting system 1500 according to the present

embodiment includes a transmission device 1510, a transmission path 120, and a non-storage type reception device 130. The transmission device 1510 is such that the transmission device 110 of the non-storage type broadcasting system 100 is additionally
5 provided with a content ID space manager 1581.

Hereinbelow, features specific to the non-storage type broadcastingsystem1500aremainlydescribed, and the descriptions of features common to the non-storage type broadcasting system 100 are omitted.

10 For each service, the content ID space manager 1501 manages the range (space) of values for the content ID of content belonging to the service, thereby designating browser content Cbs.

Forexample, the content ID space is defined by the conditions that the content is a content having the service ID "S1" if the
15 content ID is between 100 and 199; the content is a content having the service ID "S2" if the content ID is between 200 and 299; the content is a content having the service ID "S3" if the content ID is between 300 and 399, and the content is browser content Cb (Cbc) if the content ID is between 0 and 99.

20 The content ID space manager 1501 generates a content space definition CSD and outputs it to the content header adder 112. Based on the content space definition CSD, the content header adder 112 rewrites the value for the item of the content ID, which is stored in the content header Hc (HcBs) of the content body Dc (DcBs)
25 supplied from each of the content storages 113a, 113b, and 113c

via the content sender 114, into the value of a content ID adapted to the defined content ID space.

FIG. 21 shows an exemplary content space definition CSD. The content space definition CSD contains information for defining content spaces, as well as a column indicating ranges of content IDs. In this example, for each service, content is generated in which the content ID is freely set within the range from 0 to 99, and the content is stored in each content storage. Regarding service S1, the browser content Cbs generated from content body Dc supplied from the content storage 113a is outputted if it can be confirmed that the content ID thereof is between 0 and 99. In this case, if the content ID is not between 0 and 99, the browser content Cbs is considered as being an invalid content and thus removed.

For the service content Cs, which is content of the service S1, the content ID is converted into a value between 100 and 199 by adding 100 thereto. After the conversion, whether it is between 100 and 199 is confirmed. If so, the service content is outputted, and if it is not between 100 and 199, the service content is considered as being an invalid content and thus removed.

Note that during the conversion, similar conversion is performed on content IDs used for referencing other contents embedded in the content.

FIG. 22 shows an exemplary structure of the browser content Cbs after the rewriting by the content ID space manager 1501. In

the content header HcBs of the browser content Cbs, the content ID is 001 which falls within the range from 0 to 999 which is defined as the range of the content ID of the browser content Cb.

Also, the content ID space manager 1501 adds information
5 for defining a content ID space to the service attribute information stored in the service attribute information storage 111.

FIG. 20 shows exemplary service attribute information Isps. The contents of the service attribute information Isps are the same as those of the service attribute information Isp according
10 to the first embodiment shown in FIG. 4. Specifically, the operation of activating the browser content Cbs in accordance with the present embodiment is basically the same as the operation in accordance with the first embodiment described with reference to FIG. 9, except that the service attribute information Isp is changed
15 to Isps and the target content C is changed to the browser content Cbs. Accordingly, the description of the operation is omitted.

Next, referring to FIG. 23, an operation of updating the browser content Cbs by the non-storage type reception device 130 is described. As shown in FIG. 18, the operation of updating the
20 browser content Cb in accordance with the present embodiment is such that step S608 in the flowchart shown in FIG. 10 is replaced by steps S1801 and S1402.

Specifically, at step S1801, based on content C (Cn) demodulated and a content header Hc separated at step S601, a search
25 is conducted for a content C in a selected service, which has a

content ID falling within the range of the browser content Cbs designated by the service attribute information Isps. If any corresponding content C is found, the procedure proceeds to the next step S602. On the other hand, if no corresponding content
5 C is found, the procedure returns to the above-described step S601.

As described above, the non-storage type broadcasting system 1500 manages the space for content IDs, so that a content having a content ID corresponding to a specific value designated by the management can be designated as the browser content Cbs. In this
10 manner, the browser content Cbs can be designated without preparing any specific field (in this example, a column) in the content header HcBs, for example.

That is, if it is ensured that only the content provider stores the content body Dc corresponding to the service thereof
15 into the content storage 113, whatever content C is stored in the content storage 113, it is possible to prohibit the browser content Cb from being incorrectly designated for a different service and being executed in the non-storage type reception device 130.

In the present embodiment, if a single content ID space is
20 provided for all of a plurality of service providers, it is possible to predetermine the range of the content ID to be taken by the browser content Cb and thereby to prevent the browser content Cb from being incorrectly designated by the plurality of service providers. Only content ID fields may be used without preparing
25 new fields in content headers, for example, therefore it is possible

to readily use content reception-related parts of a conventional non-storage type reception device without significantly changing a conventional sending device.

While the invention has been described in detail, the foregoing description is in all aspects illustrative and not restrictive. It is understood that numerous other modifications and variations can be devised without departing from the scope of the invention.

Where an existing broadcasting infrastructure is used to implement browser transmission in the storage type broadcasting system of the present invention, it is desired or required that signal format for the broadcasting standard should be satisfied.

Reason 1: In digital broadcasting, it is prohibited to send radio waves other than those in signal formats defined by law or standards. Accordingly, introduction of a new signal format does not meet requirements of law or standards, therefore in the worst cases, the broadcaster might be deprived of the license of broadcasting.

Reason 2: The introduction of a new signal format might cause malfunction of reception devices having been connected to the broadcasting infrastructure. For example, the estimated life of a digital television set is about ten years. In this case, the requirements are not satisfied in that a currently receivable radio signal does not cause malfunction of a television set which has been shipped ten years ago.

Reason 3: The introduction of a new signal format entails significant cost in upgrading and testing broadcasting facility, and thus is not practical.

Reason 4: If requirements about an existing broadcasting standard signal format are satisfied, it is not necessary to provide an additional fixed bandwidth for browser transmission in addition to a bandwidth for transmitting signal content. That is, the same band is used for both content transmission and browser transmission. Thus, it is possible to prevent the occurrence of the waste due to the use of different bandwidths between the content and browser, e.g., the bandwidth is insufficient for content transmission, while the bandwidth for browser transmission is available. Further, it is not necessary to separately prepare a mechanism for managing the bandwidth for browser transmission.

Reason 5: If the requirements about an existing broadcasting standard signal format are satisfied, it is possible to receive a browser simultaneously with content. It is often the case that a typical conventional reception device is able to receive only one broadcast channel at a time under the constraint of cost, etc. The browser is transmitted on a channel different from that on which a television program (content) is being broadcast. Thus, a channel for transmitting the browser is received in late-night hours during which no content is being received, and browser replacement is carried out. According to the present invention, in such a case, the same broadcast channel is received under the

situation where a browser is desired at the very moment of receiving content, whereby it is possible to immediately receive the browser required for that content.

Reason 6: In the present invention, limited computation
5 resources (ROM capacity, RAM capacity, CPU processing speed) of a receiver can be effectively used regardless of a new or existing signal format. If the transmission scheme for the browser is the same as that for the content, it is meant that there is only one transmission protocol for the browser and the content.

10 In general, in order to implement a protocol stack, which is software for receiving protocols, the ROM capacity for storing programs required for processing for each protocol and the RAM capacity for the implementation are required. In the present invention, there is only one protocol, and therefore the ROM and
15 RAM capacities may be slightly greater than those of even a reception device which cannot update the browser. Further, overhead for switching between protocols is not required, therefore it is possible to simultaneously receive the browser and the content at about the same CPU processing speed.

20 Reason 7: In the present invention, the time to receive the content and the browser is completely controlled based on the time at which the browser contained in the content is transmitted from the transmission side regardless of a new or existing signal format. It is possible for the transmission side to provide control such
25 that the content is not sent out while the browser is being sent

and thereby to prevent the load on the receiver from increasing more than required. Also, by sending out a browser, which is required for presenting a specific content, at an appropriate time, it is made possible to ensure based on control by the broadcasting
5 side that a suitable browser has always been received upon receipt of the content. The control does not require the computation resources of the receiver side.

Reason 8: In the present invention, only one-way data transmission from a sending device to a reception device is required
10 for execution regardless of a new or existing signal format. Therefore, it is implementable without requiring any communication infrastructure capable of transmitting any information to the sending device via an additional communication line or the like. In the case where there are millions of reception devices, even
15 if it is necessary to replace browsers simultaneously, the sending device is only required to broadcast necessary information. Thus, the implementation is relatively easy because it is not necessary to prepare a server or a communication line for accepting browser transmission requests from millions of reception devices.

20

INDUSTRIAL APPLICABILITY

Thus, the present invention is advantageous in a non-storage type information service system which automatically provide the user in real-time with a presenter required by the user for viewing
25 a desired content with a viewer.